Sufficient Component Analysis for Supervised Dimension Reduction

Makoto Yamada YAMADA@SG.CS.TITECH.AC.JP
Gang Niu GANG@SG.CS.TITECH.AC.JP
Jun Takagi TAKAGI@SG.CS.TITECH.AC.JP
Masashi Sugiyama SUGI@CS.TITECH.AC.JP
Tokyo Institute of Technology, Tokyo 152-8552, Japan

Abstract

The purpose of sufficient dimension reduction (SDR) is to find the low-dimensional subspace of input features that is sufficient for predicting output values. In this paper, we propose a novel *distribution-free* SDR method called *sufficient component analysis* (SCA), which is computationally more efficient than existing methods. In our method, a solution is computed by iteratively performing dependence estimation and maximization: Dependence estimation is analytically carried out by recently-proposed *least-squares mutual information* (LSMI), and dependence maximization is also analytically carried out by utilizing the *Epanechnikov kernel*. Through large-scale experiments on real-world image classification and audio tagging problems, the proposed method is shown to compare favorably with existing dimension reduction approaches.

1. Introduction

The goal of *sufficient dimension reduction* (SDR) is to learn a transformation matrix $\boldsymbol{W}$ from input feature $\boldsymbol{x}$ to its low-dimensional representation $\boldsymbol{z}$ $(= \boldsymbol{W}\boldsymbol{x})$ which has 'sufficient' information for predicting output value $\boldsymbol{y}$. SDR can be formulated as the problem of finding $\boldsymbol{z}$ such that $\boldsymbol{x}$ and $\boldsymbol{y}$ are conditionally independent given $\boldsymbol{z}$ (Cook, 1998; Fukumizu et al., 2009).

Earlier SDR methods developed in statistics community, such as *sliced inverse regression* (Li, 1991), *principal Hessian direction* (Li, 1992), and *sliced average variance estimation* (Cook, 2000), rely on the elliptical assumption (e.g., Gaussian) of the data, which may not be fulfilled in practice.

To overcome the limitations of these approaches, the *kernel dimension reduction* (KDR) was proposed (Fukumizu et al., 2009). KDR employs a kernel-based dependence measure, which does not require the elliptical assumption (i.e., distribution-free), and the solution $\boldsymbol{W}$ is computed by a gradient method. Although KDR is a highly flexible SDR method, its critical weakness is the kernel function choice—the performance of KDR depends on the choice of kernel functions and the regularization parameter, but there is no systematic model selection method available. Furthermore, KDR scales poorly to massive datasets since the gradient-based optimization is computationally demanding. Another important limitation of KDR in practice is that there is no good way to set an initial solution—many random restarts may be needed for finding a good local optima, which makes the entire procedure even slower and the performance of dimension reduction unstable.

To overcome the limitations of KDR, a novel SDR method called *least-squares dimension reduction* (LSDR) was proposed recently (Suzuki & Sugiyama, 2010). LSDR adopts a squared-loss variant of mutual information as a dependency measure, which is efficiently estimated by *least-squares mutual information* (LSMI) (Suzuki et al., 2009). A notable advantage of LSDR over KDR is that kernel functions and its tuning parameters such as the kernel width and the regularization parameter can be naturally optimized by cross-validation. However, LSDR still relies on a computationally expensive gradient method and there is no good initialization scheme.

In this paper, we propose a novel SDR method called *sufficient component analysis* (SCA), which can overcome the computational inefficiency of LSDR. In SCA, the solution $\boldsymbol{W}$ in each iteration is obtained *analytically* by just solving an eigenvalue problem, which significantly contributes to improving the computational efficiency. Moreover, based on the above analytic-form

solution, we develop a method to design a good initial value for optimization, which further reduces the computational cost and help obtain a good local optimum solution.

Through large-scale experiments using the *PASCAL Visual Object Classes (VOC) 2010* dataset (Everingham et al., 2010) and the *Freesound* dataset (The Freesound Project, 2011), we demonstrate the usefulness of the proposed method.

2. Sufficient Dimension Reduction with Squared-Loss Mutual Information

In this section, we formulate the problem of *sufficient dimension reduction* (SDR) based on *squared-loss mutual information* (SMI).

2.1. Problem Formulation

Let $\mathcal{X}(\subset \mathbb{R}^d)$ be the domain of input feature $\boldsymbol{x}$ and $\mathcal{Y}$ be the domain of output data[1] $\boldsymbol{y}$. Suppose we are given n independent and identically distributed (i.i.d.) paired samples,

$$D^n = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) \mid \boldsymbol{x}_i \in \mathcal{X},\ \boldsymbol{y}_i \in \mathcal{Y},\ i = 1, \ldots, n\},$$

drawn from a joint distribution with density $p_{\mathrm{xy}}(\boldsymbol{x}, \boldsymbol{y})$.

The goal of SDR is to find a low-dimensional representation $\boldsymbol{z}$ ($\in \mathbb{R}^m$, $m \leq d$) of input $\boldsymbol{x}$ that is sufficient to describe output $\boldsymbol{y}$. More precisely, we find $\boldsymbol{z}$ such that

$$\boldsymbol{y} \perp\!\!\!\perp \boldsymbol{x} \mid \boldsymbol{z}, \tag{1}$$

meaning that, given the projected feature $\boldsymbol{z}$, the feature $\boldsymbol{x}$ is conditionally independent of output $\boldsymbol{y}$.

In this paper, we focus on linear dimension reduction scenarios:

$$\boldsymbol{z} = \boldsymbol{W}\boldsymbol{x},$$

where $\boldsymbol{W}$ is a transformation matrix. $\boldsymbol{W}$ belongs to the *Stiefel manifold* $\mathbb{S}_m^d(\mathbb{R})$:

$$\mathbb{S}_m^d(\mathbb{R}) := \{\boldsymbol{W} \in \mathbb{R}^{m \times d} | \boldsymbol{W}\boldsymbol{W}^\top = \boldsymbol{I}_m\},$$

where $^\top$ denotes the transpose and $\boldsymbol{I}_m$ is the m-dimensional identity matrix. Below, we assume that the reduced dimension m is known.

[1] $\mathcal{Y}$ could be either continuous (i.e., regression) or categorical (i.e., classification). Multi-dimensional outputs (e.g., multi-task regression and multi-label classification) and structured outputs (such as sequences, trees, and graphs) can also be handled in the proposed framework.

2.2. Dependence Estimation-Maximization Framework

Suzuki & Sugiyama (2010) showed that the optimal transformation matrix that leads to Eq.(1) can be characterized as

$$\boldsymbol{W}^* = \underset{\boldsymbol{W} \in \mathbb{R}^{m \times d}}{\operatorname{argmax}}\ \mathrm{SMI}(Z, Y)\ \text{ s.t. } \boldsymbol{W}\boldsymbol{W}^\top = \boldsymbol{I}_m. \tag{2}$$

In the above, $\mathrm{SMI}(Z, Y)$ is the *squared-loss mutual information*:

$$\mathrm{SMI}(Z, Y) := \frac{1}{2}\mathbb{E}_{p_{\mathrm{z}}, p_{\mathrm{y}}}\left[\left(\frac{p_{\mathrm{zy}}(\boldsymbol{z}, \boldsymbol{y})}{p_{\mathrm{y}}(\boldsymbol{y})p_{\mathrm{z}}(\boldsymbol{z})} - 1\right)^2\right],$$

where $\mathbb{E}_{p_{\mathrm{z}}, p_{\mathrm{y}}}$ denotes the expectation over the marginals $p_{\mathrm{z}}(\boldsymbol{z})$ and $p_{\mathrm{y}}(\boldsymbol{y})$. Note that SMI is the *Pearson divergence* from $p_{\mathrm{zy}}(\boldsymbol{z}, \boldsymbol{y})$ to $p_{\mathrm{z}}(\boldsymbol{z})p_{\mathrm{y}}(\boldsymbol{y})$, while the ordinary mutual information is the Kullback-Leibler divergence from $p_{\mathrm{zy}}(\boldsymbol{z}, \boldsymbol{y})$ to $p_{\mathrm{z}}(\boldsymbol{z})p_{\mathrm{y}}(\boldsymbol{y})$. The Pearson divergence and the Kullback-Leibler divergence both belong to the class of f-divergences, which shares similar theoretical properties. For example, SMI is non-negative and is zero if and only if Z and Y are statistically independent, as ordinary mutual information.

Based on Eq.(2), we develop the following iterative algorithm for learning $\boldsymbol{W}$:

(i) Initialization: Initialize the transformation matrix $\boldsymbol{W}$ (see Section 3.3).

(ii) Dependence estimation: For current $\boldsymbol{W}$, an SMI estimator $\widehat{\mathrm{SMI}}$ is obtained (see Section 3.1).

(iii) Dependence maximization: Given an SMI estimator $\widehat{\mathrm{SMI}}$, its maximizer with respect to $\boldsymbol{W}$ is obtained (see Section 3.2).

(iv) Convergence check: The above (ii) and (iii) are repeated until $\boldsymbol{W}$ fulfills some convergence criterion[2].

3. Proposed Method: Sufficient Component Analysis

In this section, we describe our proposed method called the *sufficient component analysis* (SCA).

3.1. Dependence Estimation

In SCA, we utilize a non-parametric SMI estimator called *least-squares mutual information* (LSMI)

[2] In experiments, we used the criterion that the improvement of $\widehat{\mathrm{SMI}}$ is less than 10^{-6}.

(Suzuki et al., 2009), which was shown to achieve the optimal convergence rate (Suzuki & Sugiyama, 2010). Here, we review LSMI.

3.1.1. Basic Idea

A key idea of LSMI is to directly estimate the *density ratio*,

$$w(\boldsymbol{z},\boldsymbol{y}) = \frac{p_{\mathrm{zy}}(\boldsymbol{z},\boldsymbol{y})}{p_{\mathrm{z}}(\boldsymbol{z})p_{\mathrm{y}}(\boldsymbol{y})},$$

without going through density estimation of $p_{\mathrm{zy}}(\boldsymbol{z},\boldsymbol{y})$, $p_{\mathrm{z}}(\boldsymbol{z})$, and $p_{\mathrm{y}}(\boldsymbol{y})$. Here, the density ratio function $w(\boldsymbol{z},\boldsymbol{y})$ is directly modeled by

$$w_{\boldsymbol{\alpha}}(\boldsymbol{z},\boldsymbol{y}) = \sum_{\ell=1}^{n} \alpha_\ell K(\boldsymbol{z},\boldsymbol{z}_\ell) L(\boldsymbol{y},\boldsymbol{y}_\ell), \tag{3}$$

where $K(\boldsymbol{z},\boldsymbol{z}')$ and $L(\boldsymbol{y},\boldsymbol{y}')$ are kernel functions for $\boldsymbol{z}$ and $\boldsymbol{y}$, respectively.

Then, the parameter $\boldsymbol{\alpha} = (\alpha_1,\dots,\alpha_n)^\top$ is learned so that the following squared error is minimized:

$$J_0(\boldsymbol{\alpha}) = \frac{1}{2}\mathbb{E}_{p_{\mathrm{z}},p_{\mathrm{y}}}\left[(w_{\boldsymbol{\alpha}}(\boldsymbol{z},\boldsymbol{y}) - w(\boldsymbol{z},\boldsymbol{y}))^2\right].$$

J_0 can be expressed as

$$J_0(\boldsymbol{\alpha}) = J(\boldsymbol{\alpha}) + \mathrm{SMI}(Z,Y) + \frac{1}{2},$$

where

$$\begin{aligned} J(\boldsymbol{\alpha}) &= \frac{1}{2}\boldsymbol{\alpha}^\top \boldsymbol{H}\boldsymbol{\alpha} - \boldsymbol{h}^\top\boldsymbol{\alpha}, \\ H_{\ell,\ell'} &= \mathbb{E}_{p_{\mathrm{z}},p_{\mathrm{y}}}\left[K(\boldsymbol{z},\boldsymbol{z}_\ell)L(\boldsymbol{y},\boldsymbol{y}_\ell)K(\boldsymbol{z},\boldsymbol{z}_{\ell'})L(\boldsymbol{y},\boldsymbol{y}_{\ell'})\right], \\ h_\ell &= \mathbb{E}_{p_{\mathrm{zy}}}\left[K(\boldsymbol{z},\boldsymbol{z}_\ell)L(\boldsymbol{y},\boldsymbol{y}_\ell)\right], \end{aligned}$$

and $\mathrm{SMI}(Z,Y)$ is constant with respect to $\boldsymbol{\alpha}$. Thus, minimizing J_0 is equivalent to minimizing J.

3.1.2. Computing the Solution

Approximating the expectations in $\boldsymbol{H}$ and $\boldsymbol{h}$ included in J by empirical averages, we arrive at the following optimization problem:

$$\min_{\boldsymbol{\alpha}}\left[\frac{1}{2}\boldsymbol{\alpha}^\top\widehat{\boldsymbol{H}}\boldsymbol{\alpha} - \widehat{\boldsymbol{h}}^\top\boldsymbol{\alpha} + \lambda\boldsymbol{\alpha}^\top\boldsymbol{R}\boldsymbol{\alpha}\right],$$

where a regularization term $\lambda\boldsymbol{\alpha}^\top\boldsymbol{R}\boldsymbol{\alpha}$ is included for avoiding overfitting, $\lambda\ (\geq 0)$ is a regularization parameter, $\boldsymbol{R}$ is a regularization matrix, and, for $\boldsymbol{z}_i = \boldsymbol{W}\boldsymbol{x}_i$,

$$\begin{aligned} \widehat{H}_{\ell,\ell'} &= \frac{1}{n^2}\sum_{i,j=1}^{n} K(\boldsymbol{z}_i,\boldsymbol{z}_\ell)L(\boldsymbol{y}_i,\boldsymbol{y}_\ell)K(\boldsymbol{z}_j,\boldsymbol{z}_{\ell'})L(\boldsymbol{y}_j,\boldsymbol{y}_{\ell'}), \\ \widehat{h}_\ell &= \frac{1}{n}\sum_{i=1}^{n} K(\boldsymbol{z}_i,\boldsymbol{z}_\ell)L(\boldsymbol{y}_i,\boldsymbol{y}_\ell). \end{aligned}$$

Differentiating the above objective function with respect to $\boldsymbol{\alpha}$ and equating it to zero, we can obtain an analytic-form solution:

$$\widehat{\boldsymbol{\alpha}} = (\widehat{\boldsymbol{H}} + \lambda\boldsymbol{R})^{-1}\widehat{\boldsymbol{h}}. \tag{4}$$

Based on the fact that $\mathrm{SMI}(Z,Y)$ is expressed as

$$\mathrm{SMI}(Z,Y) = \frac{1}{2}\mathbb{E}_{p_{\mathrm{zy}}}\left[w(\boldsymbol{z},\boldsymbol{y})\right] - \frac{1}{2},$$

the following SMI estimator can be obtained:

$$\widehat{\mathrm{SMI}} = \frac{1}{2}\widehat{\boldsymbol{h}}^\top\widehat{\boldsymbol{\alpha}} - \frac{1}{2}. \tag{5}$$

3.1.3. Model Selection

Hyper-parameters included in the kernel functions and the regularization parameter can be optimized by cross-validation with respect to J.

More specifically, the samples $\mathcal{Z} = \{(\boldsymbol{z}_i,\boldsymbol{y}_i)\}_{i=1}^n$ are divided into K disjoint subsets $\{\mathcal{Z}_k\}_{k=1}^K$ of (approximately) the same size. Then, an estimator $\widehat{\boldsymbol{\alpha}}_{\mathcal{Z}_k}$ is obtained using $\mathcal{Z}\backslash\mathcal{Z}_k$ (i.e,. all samples without $\mathcal{Z}_k$), and the approximation error for the hold-out samples $\mathcal{Z}_k$ is computed as

$$J_{\mathcal{Z}_k}^{(K\text{-CV})} = \frac{1}{2}\widehat{\boldsymbol{\alpha}}_{\mathcal{Z}_k}^\top\widehat{\boldsymbol{H}}_{\mathcal{Z}_k}\widehat{\boldsymbol{\alpha}}_{\mathcal{Z}_k} - \widehat{\boldsymbol{h}}_{\mathcal{Z}_k}^\top\widehat{\boldsymbol{\alpha}}_{\mathcal{Z}_k},$$

where, for $|\mathcal{Z}_k|$ being the number of samples in the subset $\mathcal{Z}_k$,

$$\begin{aligned} [\widehat{H}_{\mathcal{Z}_k}]_{\ell,\ell'} &= \frac{1}{|\mathcal{Z}_k|^2}\sum_{(\boldsymbol{z},\boldsymbol{y}),(\boldsymbol{z}',\boldsymbol{y}')\in\mathcal{Z}_k} K(\boldsymbol{z},\boldsymbol{z}_\ell)L(\boldsymbol{y},\boldsymbol{y}_\ell) \\ &\qquad \times K(\boldsymbol{z}',\boldsymbol{z}_{\ell'})L(\boldsymbol{y}',\boldsymbol{y}_{\ell'}), \\ [\widehat{h}_{\mathcal{Z}_k}]_\ell &= \frac{1}{|\mathcal{Z}_k|}\sum_{(\boldsymbol{z},\boldsymbol{y})\in\mathcal{Z}_k} K(\boldsymbol{z},\boldsymbol{z}_\ell)L(\boldsymbol{y},\boldsymbol{y}_\ell). \end{aligned}$$

This procedure is repeated for $k = 1,\dots,K$, and its average $J^{(K\text{-CV})}$ is outputted as

$$J^{(K\text{-CV})} = \frac{1}{K}\sum_{k=1}^{K} J_{\mathcal{Z}_k}^{(K\text{-CV})}.$$

We compute $J^{(K\text{-CV})}$ for all model candidates, and choose the model that minimizes $J^{(K\text{-CV})}$.

3.2. Dependence Maximization

Given an SMI estimator $\widehat{\mathrm{SMI}}$ (5), we next show how $\widehat{\mathrm{SMI}}$ can be efficiently maximized with respect to $\boldsymbol{W}$:

$$\max_{\boldsymbol{W}\in\mathbb{R}^{m\times d}} \widehat{\mathrm{SMI}} \quad \text{s.t. } \boldsymbol{W}\boldsymbol{W}^\top = \boldsymbol{I}_m.$$

We propose to use a truncated negative quadratic function called the *Epanechnikov kernel* (Epanechnikov, 1969) as a kernel for $\boldsymbol{z}$:

$$K(\boldsymbol{z}, \boldsymbol{z}_\ell) = \max\left(0, 1 - \frac{\|\boldsymbol{z} - \boldsymbol{z}_\ell\|^2}{2\sigma_{\mathrm{z}}^2}\right).$$

Let $I(c)$ be the indicator function, i.e., $I(c) = 1$ if c is true and zero otherwise. Then, for the above kernel, $\widehat{\mathrm{SMI}}$ can be expressed as

$$\widehat{\mathrm{SMI}} = \frac{1}{2}\mathrm{tr}\left(\boldsymbol{W}\boldsymbol{D}\boldsymbol{W}^\top\right) - \frac{1}{2},$$

where $\mathrm{tr}(\boldsymbol{A})$ is the trace of matrix $\boldsymbol{A}$, and

$$\boldsymbol{D} = \frac{1}{n}\sum_{i=1}^{n}\sum_{\ell=1}^{n}\widehat{\alpha}_\ell(\boldsymbol{W})I\left(\frac{\|\boldsymbol{W}\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{x}_\ell\|^2}{2\sigma_{\mathrm{z}}^2} < 1\right) \times L(\boldsymbol{y}_i, \boldsymbol{y}_\ell)\left[\frac{1}{m}\boldsymbol{I}_d - \frac{1}{2\sigma_{\mathrm{z}}^2}(\boldsymbol{x}_i - \boldsymbol{x}_\ell)(\boldsymbol{x}_i - \boldsymbol{x}_\ell)^\top\right].$$

Here, by $\widehat{\alpha}_\ell(\boldsymbol{W})$, we explicitly indicated the fact that $\widehat{\alpha}_\ell$ depends on $\boldsymbol{W}$.

Let $\boldsymbol{D}'$ be $\boldsymbol{D}$ with $\boldsymbol{W}$ replaced by $\boldsymbol{W}'$, where $\boldsymbol{W}'$ is a transformation matrix obtained in the previous iteration. Thus, $\boldsymbol{D}'$ no longer depends on $\boldsymbol{W}$. Here we replace $\boldsymbol{D}$ in $\widehat{\mathrm{SMI}}$ by $\boldsymbol{D}'$, which gives the following simplified SMI estimate:

$$\frac{1}{2}\mathrm{tr}\left(\boldsymbol{W}\boldsymbol{D}'\boldsymbol{W}^\top\right) - \frac{1}{2}. \tag{6}$$

A maximizer of Eq.(6) can be analytically obtained by $(\boldsymbol{w}_1|\cdots|\boldsymbol{w}_m)^\top$, where $\{\boldsymbol{w}_i\}_{i=1}^m$ are the m principal components of $\boldsymbol{D}'$.

3.3. Initialization of $\boldsymbol{W}$

In the dependence estimation-maximization framework described in Section 2.2, initialization of the transformation matrix $\boldsymbol{W}$ is important. Here we propose to initialize it based on dependence maximization without dimensionality reduction.

More specifically, we determine the initial transformation matrix as $(\boldsymbol{w}_1^{(0)}|\cdots|\boldsymbol{w}_m^{(0)})^\top$, where $\{\boldsymbol{w}_i^{(0)}\}_{i=1}^m$ are the m principal components of $\boldsymbol{D}^{(0)}$:

$$\begin{aligned}
\boldsymbol{D}^{(0)} &= \frac{1}{n}\sum_{i=1}^{n}\sum_{\ell=1}^{n}\widehat{\alpha}_\ell^{(0)}I\left(\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_\ell\|^2}{2\sigma_{\mathrm{x}}^2} < 1\right)L(\boldsymbol{y}_i, \boldsymbol{y}_\ell) \\
&\quad \times \left[\frac{1}{m}\boldsymbol{I}_d - \frac{1}{2\sigma_{\mathrm{x}}^2}(\boldsymbol{x}_i - \boldsymbol{x}_\ell)(\boldsymbol{x}_i - \boldsymbol{x}_\ell)^\top\right], \\
\widehat{\boldsymbol{\alpha}}^{(0)} &= (\widehat{\boldsymbol{H}}^{(0)} + \lambda\boldsymbol{R})^{-1}\widehat{\boldsymbol{h}}^{(0)}, \\
\widehat{H}_{\ell,\ell'}^{(0)} &= \frac{1}{n^2}\sum_{i,j=1}^{n}K'(\boldsymbol{x}_i, \boldsymbol{x}_\ell)L(\boldsymbol{y}_i, \boldsymbol{y}_\ell) \\
&\quad \times K'(\boldsymbol{x}_j, \boldsymbol{x}_{\ell'})L(\boldsymbol{y}_j, \boldsymbol{y}_{\ell'}), \\
\widehat{h}_\ell^{(0)} &= \frac{1}{n}\sum_{i=1}^{n}K'(\boldsymbol{x}_i, \boldsymbol{x}_\ell)L(\boldsymbol{y}_i, \boldsymbol{y}_\ell), \\
K'(\boldsymbol{x}, \boldsymbol{x}_\ell) &= \max\left(0, 1 - \frac{\|\boldsymbol{x} - \boldsymbol{x}_\ell\|^2}{2\sigma_{\mathrm{x}}^2}\right).
\end{aligned}$$

σ_{x} is the kernel width and is chosen by cross-validation (see Section 3.1.3).

4. Relation to Existing Methods

Here, we review existing SDR methods and discuss the relation to the proposed SCA method.

4.1. Kernel Dimension Reduction

Kernel Dimension Reduction (KDR) (Fukumizu et al., 2009) tries to directly maximize the conditional independence of $\boldsymbol{x}$ and $\boldsymbol{y}$ given $\boldsymbol{z}$ under a kernel-based independence measure.

The KDR learning criterion is given by

$$\begin{aligned}
\boldsymbol{W}^* = &\operatorname*{argmax}_{\boldsymbol{W}\in\mathbb{R}^{m\times d}}\mathrm{tr}\left[\widetilde{\boldsymbol{L}}(\widetilde{\boldsymbol{K}} + n\epsilon\boldsymbol{I}_n)^{-1}\right] \\
&\text{s.t. } \boldsymbol{W}\boldsymbol{W}^\top = \boldsymbol{I}_m,
\end{aligned} \tag{7}$$

where $\widetilde{\boldsymbol{L}} = \boldsymbol{\Gamma}\boldsymbol{L}\boldsymbol{\Gamma}$, $\boldsymbol{\Gamma} = \boldsymbol{I} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$, $L_{i,j} = L(\boldsymbol{y}_i, \boldsymbol{y}_j)$, $\widetilde{\boldsymbol{K}} = \boldsymbol{\Gamma}\boldsymbol{K}\boldsymbol{\Gamma}$, $K_{i,j} = K(\boldsymbol{z}_i, \boldsymbol{z}_j)$, and ϵ is a regularization parameter.

Solving the above optimization problem is cumbersome since the objective function is non-convex. In the original KDR paper (Fukumizu et al., 2009), a gradient method is employed for finding a local optimal solution. However, the gradient-based optimization is computationally demanding due to its slow convergence and it requires many restarts for finding a good local optima. Thus, KDR scales poorly to massive datasets.

Another critical weakness of KDR is the kernel function choice. The performance of KDR depends on the choice of kernel functions and the regularization parameter, but there is no systematic model selection

method for KDR available. Using the Gaussian kernel with its width set to the median distance between samples is a standard heuristic in practice, but this does not always work very well.

Furthermore, KDR lacks a good way to set an initial solution in the gradient procedure. Then, in practice, we need to run the algorithm many times with random initial points for finding good local optima. However, this makes the entire procedure even slower and the performance of dimension reduction unstable.

The proposed SCA method can successfully overcome the above weaknesses of KDR—SCA is equipped with cross-validation for model selection (Section 3.1.3), its solution can be computed analytically (see Section 3.2), and a systematic initialization scheme is available (see Section 3.3).

4.2. Least-Squares Dimensionality Reduction

Least-squares dimension reduction (LSDR) is a recently proposed SDR method that can overcome the limitations of KDR (Suzuki & Sugiyama, 2010). That is, LSDR is equipped with a natural model selection procedure based on cross-validation.

The proposed SCA can actually be regarded as a computationally efficient alternative to LSDR. Indeed, LSDR can also be interpreted as a dependence estimation-maximization algorithm (see Section 2.2), and the dependence estimation procedure is essentially the same as the proposed SCA, i.e., LSMI is used. The dependence maximization procedure is different from SCA—LSDR uses a *natural gradient* method (Amari, 1998).

In LSDR, the following SMI estimator is used:

$$\widehat{\mathrm{SMI}} = \widehat{\boldsymbol{\alpha}}^\top \widehat{\boldsymbol{h}} - \frac{1}{2}\widehat{\boldsymbol{\alpha}}^\top \widehat{\boldsymbol{H}}\widehat{\boldsymbol{\alpha}} - \frac{1}{2},$$

where $\widehat{\boldsymbol{\alpha}}$, $\widehat{\boldsymbol{h}}$ and $\widehat{\boldsymbol{H}}$ are defined in Section 3.1. Then the gradient of $\widehat{\mathrm{SMI}}$ is given by

$$\frac{\partial \widehat{\mathrm{SMI}}}{\partial W_{\ell,\ell'}} = \frac{\partial \widehat{\boldsymbol{h}}^\top}{\partial W_{\ell,\ell'}}(2\widehat{\boldsymbol{\alpha}} - \widehat{\boldsymbol{\beta}}) - \widehat{\boldsymbol{\alpha}}^\top \frac{\partial \widehat{\boldsymbol{H}}}{\partial W_{\ell,\ell'}}(\frac{3}{2}\widehat{\boldsymbol{\alpha}} - \widehat{\boldsymbol{\beta}}) + \widehat{\boldsymbol{\alpha}}^\top \frac{\partial \boldsymbol{R}}{\partial W_{\ell,\ell'}}(\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\alpha}}),$$

where $\widehat{\boldsymbol{\beta}} = (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})^{-1}\widehat{\boldsymbol{H}}\widehat{\boldsymbol{\alpha}}$. The *natural gradient* update of $\boldsymbol{W}$, which takes into account the structure of the Stiefel manifold (Amari, 1998), is given by

$$\boldsymbol{W} \leftarrow \boldsymbol{W} \exp\left(\eta\left(\boldsymbol{W}^\top \frac{\partial \widehat{\mathrm{SMI}}}{\partial \boldsymbol{W}} - \frac{\partial \widehat{\mathrm{SMI}}}{\partial \boldsymbol{W}}^\top \boldsymbol{W}\right)\right),$$

where 'exp' for a matrix denotes the *matrix exponential*. $\eta \geq 0$ is a step size, which may be optimized by a line-search method such as *Armijo's rule* (Patriksson, 1999).

Since cross-validation is available for model selection of LSMI, LSDR is more favorable than KDR. However, its optimization still relies on a gradient-based method and thus it is computationally expensive.

Furthermore, there seems no good initialization scheme of the transformation matrix $\boldsymbol{W}$. In the original paper by Suzuki & Sugiyama (2010), initial values were chosen randomly and the gradient method was run many times for finding a better local solution.

The proposed SCA method can successfully overcome the above weaknesses of LSDR, by providing an analytic-form solution (see Section 3.2) and a systematic initialization scheme (see Section 3.3).

5. Experiments

In this section, we experimentally investigate the performance of the proposed and existing SDR methods using artificial and real-world datasets.

5.1. Artificial Datasets

We use four artificial datasets, and compare the proposed SCA, LSDR[1] (Suzuki & Sugiyama, 2010), KDR[2] (Fukumizu et al., 2009), sliced inverse regression (SIR)[3] (Li, 1991), sliced average variance estimation (SAVE)[3] (Cook, 2000), and principal Hessian direction (pHd)[3] (Li, 1992).

In SCA, we use the Gaussian kernel for $\boldsymbol{y}$:

$$L(\boldsymbol{y}, \boldsymbol{y}_\ell) = \exp\left(-\frac{\|\boldsymbol{y} - \boldsymbol{y}_\ell\|^2}{2\sigma_{\mathrm{y}}}\right).$$

The identity matrix is used as regularization matrix $\boldsymbol{R}$, and the kernel widths σ_{x}, σ_{y}, and σ_{z} as well as the regularization parameter λ are chosen based on 5-fold cross-validation.

The performance of each method is measured by

$$\frac{1}{\sqrt{2m}}\|\widehat{\boldsymbol{W}}^\top \widehat{\boldsymbol{W}} - \boldsymbol{W}^{*\top}\boldsymbol{W}^*\|_{\mathrm{Frobenius}},$$

where $\|\cdot\|_{\mathrm{Frobenius}}$ denotes the Frobenius norm, $\widehat{\boldsymbol{W}}$ is an estimated transformation matrix, and $\boldsymbol{W}^*$ is the

[1] http://sugiyama-www.cs.titech.ac.jp/~sugi/software/LSDR/index.html

[2] We used the program code provided by one of the authors of Fukumizu et al. (2009), which 'anneals' the Gaussian kernel width over gradient iterations.

[3] http://mirrors.dotsrc.org/cran/web/packages/dr/index.html

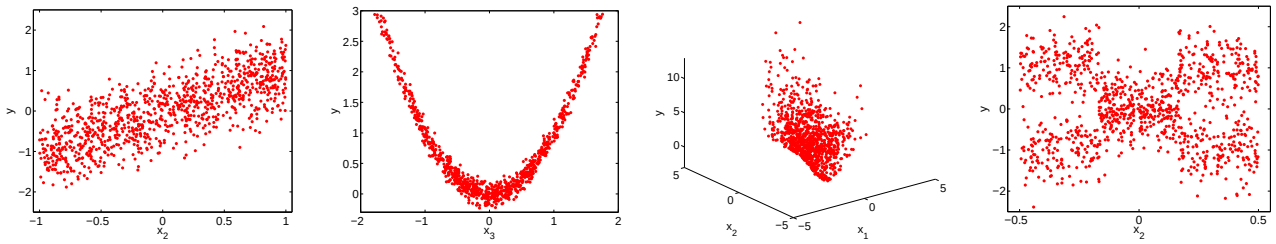


Figure 1. Artificial datasets.

Table 1. Mean of Frobenius-norm error (with standard deviations in brackets) and mean CPU time over 100 trials. Computation time is normalized so that LSDR is one. LSDR was repeated 5 times with random initialization and the transformation matrix with the minimum CV score was chosen as the final solution. 'SCA(0)' indicates the performance of the initial transformation matrix obtained by the method described in Section 3.3. The best method in terms of the mean Frobenius-norm and comparable methods according to the *t-test* at the significance level 1% are specified by bold face.

Datasets	d	m	SCA(0)	SCA	LSDR	KDR	SIR	SAVE	pHd
Data1	4	1	.089(.042)	**.048(.031)**	**.056 (.021)**	**.048(.019)**	.257 (.168)	.339 (.218)	.593 (.210)
Data2	10	1	.078(.019)	**.007(.002)**	.039 (.023)	.024 (.007)	.431 (.281)	.348 (.206)	.443 (.222)
Data3	4	2	.065(.035)	**.018(.010)**	.090 (.069)	**.029(.119)**	.362 (.182)	.343 (.213)	.437 (.231)
Data4	5	1	.118(.046)	**.042(.030)**	.151 (.296)	.118 (.238)	.421 (.268)	.356 (.197)	.591 (.205)
Time			0.03	0.49	1.0	0.96	<0.01	<0.01	<0.01

optimal transformation matrix. Note that the above error measure takes its value in $[0, 1]$.

We use the following four datasets (see Figure 1):

(a) Data1:

$$Y = X_2 + 0.5E,$$

where $(X_1, \ldots, X_4)^\top \sim U([-1\ 1]^4)$ and $E \sim N(0, 1)$. Here, $U(\mathcal{S})$ denotes the uniform distribution on $\mathcal{S}$, and $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the Gaussian distribution with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$.

(b) Data2:

$$Y = (X_3)^2 + 0.1E,$$

where $(X_1, \ldots, X_{10})^\top \sim N(\mathbf{0}_{10}, \boldsymbol{I}_{10})$ and $E \sim N(0, 1)$.

(c) Data3:

$$Y = \frac{(X_1)^2 + X_2}{0.5 + (X_2 + 1.5)^2} + (1 + X_2)^2 + 0.1E,$$

where $(X_1, \ldots, X_4)^\top \sim N(\mathbf{0}_4, \boldsymbol{I}_4)$ and $E \sim N(0, 1)$.

(d) Data4:

$$Y|X_2 \sim \begin{cases} N(0, 0.2) & \text{if } X_2 \leq |1/6| \\ 0.5N(1, 0.2) & \text{otherwise} \\ +0.5N(-1, 0.2), & \end{cases}$$

where $(X_1, \ldots, X_5)^\top \sim U([-0.5\ 0.5]^5)$ and $E \sim N(0, 1)$.

The performance of each method is summarized in Table 1, which depicts the mean and standard deviation of the Frobenius-norm error over 100 trials when the number of samples is $n = 1000$. As can be observed, the proposed SCA overall performs well. 'SCA(0)' in the table indicates the performance of the initial transformation matrix obtained by the method described in Section 3.3. The result shows that SCA(0) gives a reasonably good transformation matrix with a tiny computational cost. Note that KDR and LSDR have high standard deviation for Data3 and Data4, meaning that KDR and LSDR sometimes perform poorly.

5.2. Multi-label Classification for Real-world Datasets

Finally, we evaluate the performance of the proposed method in real-world multi-label classification problems.

5.2.1. Setup

Below, we compare SCA, Multi-label Dimensionality reduction via Dependence Maximization (MDDM)[4] (Zhang & Zhou, 2010), Canonical Correlation Anal-

[4] http://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/annex/MDDM.htm

ysis (CCA)[5] (Hotelling, 1936), and Principal Component Analysis (PCA)[6] (Bishop, 2006). We use a real-world image classification dataset called the *PASCAL Visual Object Classes (VOC) 2010* dataset (Everingham et al., 2010) and a real-world automatic audio-tagging dataset called the *Freesound* dataset (The Freesound Project, 2011). Since the computational costs of KDR and LSDR were unbearably large, we decided not to include them in the comparison.

We employ the misclassification rate by the nearest-neighbor classifier as a performance measure:

$$\mathrm{err} = \frac{1}{nc}\sum_{i=1}^{n}\sum_{k=1}^{c} I(\widehat{y}_{i,k} \neq y_{i,k}),$$

where c is the number of classes, $\widehat{y}$ and y are the estimated and true labels, and $I(y \neq y')$ is the indicator function.

For SCA and MDDM, we use the following kernel function (Sarwar et al., 2001) for $\boldsymbol{y}$:

$$L(\boldsymbol{y}, \boldsymbol{y}') = \frac{(\boldsymbol{y}-\overline{\boldsymbol{y}})^\top(\boldsymbol{y}'-\overline{\boldsymbol{y}})}{\|\boldsymbol{y}-\overline{\boldsymbol{y}}\|\|\boldsymbol{y}'-\overline{\boldsymbol{y}}'\|},$$

where $\overline{\boldsymbol{y}}$ is the sample mean: $\overline{\boldsymbol{y}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{y}_i$.

5.2.2. PASCAL VOC 2010 Dataset

The VOC 2010 dataset consists of 20 binary classification tasks of identifying the existence of a person, aeroplane, etc. in each image. The total number of images in the dataset is 11319, and we used 1000 randomly chosen images for training and the rest for testing.

In this experiment, we first extracted visual features from each image using the *Speed Up Robust Features* (SURF) algorithm (Bay et al., 2008), and obtained 500 *visual words* as the cluster centers in the SURF space. Then, we computed a 500-dimensional *bag-of-feature* vector by counting the number of visual words in each image. We randomly sampled the training and test data 100 times, and computed the means and standard deviations of the classification error.

The results are plotted in Figure 2(a), showing that SCA outperforms the existing methods, and SCA is the only method that outperforms 'ORI' (no dimension reduction)—SCA achieves almost the same error rate as 'ORI' with only a 10-dimensional subspace.

5.2.3. Freesound Dataset

The *Freesound* dataset (The Freesound Project, 2011) consists of various audio files annotated with word tags such as 'people', 'noisy', and 'restaurant'. We used 230 tags in this experiment. The total number of audio files in the dataset is 5905, and we used 1000 randomly chosen audio files for training and the rest for testing.

We first extracted *Mel-Frequency Cepstrum Coefficients* (MFCC) (Rabiner & Juang, 1993) from each audio file, and obtained 1024 *audio features* as the cluster centers in MFCC. Then, we computed a 1024-dimensional *bag-of-feature* vector by counting the number of audio features in each audio file. We randomly chose the training and test samples 100 times, and computed the means and standard deviations of the classification error.

The results plotted in Figure 2(b) show that, similarly to the image classification task, the proposed SCA outperforms the existing methods, and SCA is the only method that outperforms 'ORI'.

6. Conclusion

In this paper, we proposed a novel *sufficient dimension reduction* (SDR) method called *sufficient component analysis* (SCA), which is computationally more efficient than existing SDR methods. In SCA, a transformation matrix was estimated by iteratively performing dependence estimation and maximization, both of which are *analytically* carried out. Moreover, we developed a systematic method to design a good initial transformation matrix, which highly contributes to further reducing the computational cost and help obtain a good local optimum solution. We applied the proposed SCA to real-world image classification and audio tagging tasks, and experimentally showed that the proposed method is promising.

Acknowledgments

The authors thank Prof. Kenji Fukumizu for providing us the KDR code and Prof. Taiji Suzuki for his valuable comments. MY was supported by the JST PRESTO program. GN was supported by the MEXT scholarship. MS was supported by SCAT, AOARD, and the JST PRESTO program.

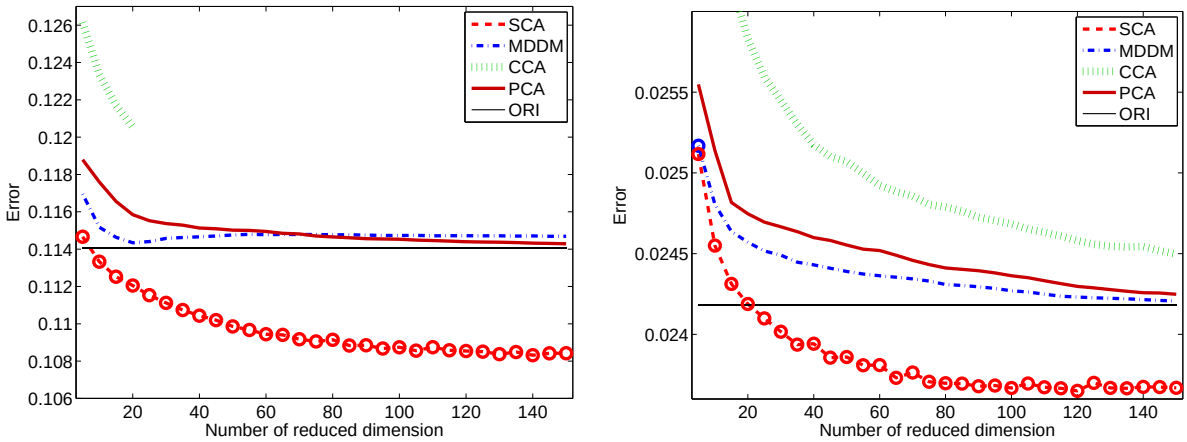
References

Amari, S. Natural gradient works efficiently in learning. *Neural Computation*, 10:251–276, 1998.

Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

Bishop, C. M. *Pattern Recognition and Machine*

[5] http://www.mathworks.com/help/toolbox/stats/canoncorr.html
[6] http://www.mathworks.com/help/toolbox/stats/princomp.html



(a) VOC 2010 dataset

(b) Freesound dataset

Figure 2. Results on image classification with VOC 2010 dataset and audio classification with Freesound datasets. Misclassification rate when the one-nearest-neighbor classifier is used as a classifier is reported. The best dimension reduction method in terms of the mean error and comparable methods according to the t-test at the significance level 1% are specified by '○'. CCA can be applied to dimension reduction up to c dimensions, where c is the number of classes ($c = 20$ in VOC 2010 and $c = 230$ in Freesound). 'ORI' denotes the original data without dimension reduction.

Learning. Springer, New York, NY, 2006.

Cook, R. D. *Regression graphics: Ideas for studying regressions through graphics*. Wiley, New York, 1998.

Cook, R. D. Save: A method for dimension reduction and graphics in regression. *Theory and Methods*, 29: 2109–2121, 2000.

Epanechnikov, V. Nonparametric estimates of a multivariate probability density. *Theory of Probability and its Applications*, 14:153–158, 1969.

Everingham, M., Gool, L. V., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html, 2010.

Fukumizu, K., Bach, F. R., and Jordan, M. Kernel dimension reduction in regression. *The Annals of Statistics*, 37(4):1871–1905, 2009.

Hotelling, H. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.

Li, K.-C. Sliced inverse regression for dimension reduction. *Journal of American Statistical Association*, 86:316–342, 1991.

Li, K.-C. On principal Hessian directions for data visualization and dimension reduction: Another application of Steinfs lemma. *Journal of American Statistical Association*, 87:1025–1034, 1992.

Patriksson, M. *Nonlinear Programming and Variational Inequality Problems*. Kluwer Academic, Dredrecht, 1999.

Rabiner, L. and Juang, B-H. *Fundamentals of Speech Recognition*. Prentice Hall, Englewood Cliffs, NJ, 1993.

Sarwar, B., Karypis, G., Konstan, J., and Reidl, J. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web (WWW2001)*, pp. 285–295, 2001.

Suzuki, T. and Sugiyama, M. Sufficient dimension reduction via squared-loss mutual information estimation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS2010)*, pp. 804–811, 2010.

Suzuki, T., Sugiyama, M., Kanamori, T., and Sese, J. Mutual information estimation reveals global associations between stimuli and biological processes. *BMC Bioinformatics*, 10(S52), 2009.

The Freesound Project. Freesound, 2011. http://www.freesound.org.

Zhang, Y. and Zhou, Z.-H. Multilabel dimensionality reduction via dependence maximization. *ACM Trans. Knowl. Discov. Data*, 4:14:1–14:21, 2010. ISSN 1556-4681.